



An Approach to Dynamical Distance Geometry

Antonio Mucherino, Douglas S. Gonçalves

► To cite this version:

Antonio Mucherino, Douglas S. Gonçalves. An Approach to Dynamical Distance Geometry. GSI 2017 - International Conference on Geometric Science of Information, Nov 2017, Paris, France. pp.821-829, 10.1007/978-3-319-68445-1_94 . hal-01632847

HAL Id: hal-01632847

<https://inria.hal.science/hal-01632847>

Submitted on 7 Dec 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An Approach to Dynamical Distance Geometry

Antonio Mucherino¹, Douglas S. Gonçalves²

¹ IRISA, University of Rennes 1, Rennes, France.

Email: antonio.mucherino@irisa.fr

² Department of Mathematics, CFM, Federal University of Santa Catarina,
Florianópolis, Brazil. Email: douglas@mtm.ufsc.br

Abstract. We introduce the dynamical distance geometry problem (dynDGP), where vertices of a given simple weighted undirected graph are to be embedded at different times t . Solutions to the dynDGP can be seen as motions of a given set of objects. In this work, we focus our attention on a class of instances where motion inter-frame distances are not available, and reduce the problem of embedding every motion frame as a static distance geometry problem. Some preliminary computational experiments are presented.

1 Introduction

Given a simple weighted undirected graph G , the Distance Geometry Problem (DGP) asks whether there exists an embedding of its vertex set into a Euclidean space \mathbb{R}^K so that the distances between embedded vertices correspond to the weights assigned to the edges of G [10]. There is a growing interest in this problem, as it is shown by the increasing number of books and journal collections on this topic (see for example [11, 12], other publications are currently under production). However, in most of the published material on this topic, the DGP is presented as a static problem, and there is only a little mention to its potential extension to dynamical applications. This paper presents a preliminary step toward the study of dynamical DGPs.

We focus our attention on the class of problems where the instances can be represented by employing a graph G whose vertex set $V \times T$ is the product of two sets: a set V representing predefined objects, and a set $T \subset \mathbb{N}$ of temporal values. The vertex $\{v, t\} \in V \times T$ represents a given object v at a certain instant t (in the following, we will use the notation v_t for the vertex $\{v, t\}$ of G). The edge set E contains edges $\{u_q, v_t\}$, whose weights provide information about the distance between two vertices u and v at times q and t , respectively. In this context, a possible embedding for G is one of the possible *motions* for the objects in V that are compatible with the distance constraints in G .

Let $G = (V \times T, E, d)$ be therefore a simple weighted undirected graph representing an instance of the *dynamical* DGP (dynDGP). The function

$$d : \{u_q, v_t\} \in E \longrightarrow (\delta, \pi) \in \mathbb{R}_+ \times \mathbb{R}_+,$$

assigns pairs of nonnegative real values to every pair of vertices of $V \times T$ belonging to the edge set of G . The nonnegative value δ is the distance between u and v at times q

and t , respectively, while π is a nonnegative value representing the “importance” of the distances δ (higher values indicate higher importance). We also refer to π as the *priority level* of the distance δ .

From a given instance of the dynDGP, we can extract some meaningful sub-instances. For example, the subgraph $G[\{v\} \times T]$ corresponds to the trajectory of one v_t for different times t . Moreover, the subgraph $G[V \times \{t\}]$, induced by the set product between V and only one temporal value, gives a “classical” DGP instance for a fixed time t . In this context, a possible embedding of $G[V \times \{t\}]$ is a candidate *frame* at time t of a motion which is an embedding for G . Let E_t be the edge set of the induced subgraph $G[V \times \{t\}]$. The set $\hat{E} = E \setminus \bigcup_{t \in T} E_t$ contains all edges that relate vertices at different times t . In this

work, we will make the assumption that $\hat{E} = \emptyset$. As a consequence, with a little abuse of notation, we will use the notation δ_{uv}^t for indicating the distance between u and v at time t . We will use a similar notation π_{uv}^t for the priorities of such distances. We will omit the time t when not relevant.

Our approach to this class of the dynDGP is to solve, for every time $t \in T$, a classical DGP on the sub-instance $G[V \times \{t\}]$. For solving the sub-instances $G[V \times \{t\}]$, we consider the optimization problem proposed in [6], which solves DGPs where priorities are associated to the available distances. We tackle this optimization problem with a non-monotone spectral gradient method [1, 2, 16], and obtain a fluid motion even in absence of distances in the subset \hat{E} .

The rest of the paper is organized as follows. We will provide the details of the optimization problem to be solved for every sub-instance $G[V \times \{t\}]$ in Section 2, while Section 3 will present the nonmonotone spectral gradient method. In Section 4, we will propose some computational experiments on a set of motions from which we derived instances of the dynDGP such that $\hat{E} = \emptyset$. Finally, Section 5 concludes the paper with a short discussion on some applications that can be modeled by a dynDGP.

2 An optimization problem with priorities on distances

Let $G = (V \times T, E, d)$ be an instance of the dynDGP such that $\hat{E} = \emptyset$. In these hypotheses, there are no distances in E that relate vertices u_q and v_t such that the times q and t are different (see Introduction). Therefore, we can split our dynDGP instance in a sequence of static DGPs represented by the subgraphs $G[V \times \{t\}]$, for each time t , without losing any information. In order to guarantee a smooth variation for the positions assigned to the vertices v_t , we consider, for increasing time values t , the solution found when solving the sub-instance $G[V \times \{t-1\}]$ as an initial approximation for the sub-instance $G[V \times \{t\}]$.

As already described in the Introduction, we suppose that our graphs G have two kinds of weights assigned to the edges. The weight δ_{uv}^t is a numerical approximation of the distance between the vertices u_t and v_t at the same time t ; the weight π_{uv}^t represents the priority on the distance δ_{uv}^t . In fact, since our distances δ_{uv}^t may be not precise, and we have no a priori information on the error they may carry, it is fundamental to coupled them with the priority levels π_{uv}^t . This way, even if the distance matrix $[\delta_{uv}]$ is not a Euclidean distance matrix (EDM), we can seek an embedding where distances

having a higher priority are privileged. This approach is also justified by some of the potential applications of the dynDGP (see Conclusions).

For solving every sub-instance $G[V \times \{t\}]$, we consider the optimization problem proposed by Glunt et al. in [6]. Let $n = |V \times \{t\}| = |V|$ and recall that E_t is the edge set of the induced subgraph $G[V \times \{t\}]$. At each time t , we seek a set of positions $\{x_1^t, x_2^t, \dots, x_n^t\} \in \mathbb{R}^K$ for the vertices in $V \times \{t\}$, which minimizes the following objective function:

$$\sigma(X) = \frac{1}{2} \sum_{\{u,v\} \in E_t} \pi_{uv} (\|x_u - x_v\| - \delta_{uv})^2,$$

where $X = [x_1 \ x_2 \ \dots \ x_n]^T \in \mathbb{R}^{n \times K}$ is a matrix representation of the set of positions for the vertices, with x_v a column vector. Notice that we omitted the time t because it is supposed to be fixed in every subgraph. The variables defined in the following and derived from X are supposed to inherit the same matrix structure.

As shown in [5], the function $\sigma(X)$ is differentiable at X if and only if $\|x_u - x_v\| > 0$ for all $\{u, v\} \in E_t$ such that $\pi_{uv} \delta_{uv} > 0$. In such a case, the gradient can be written as $\nabla \sigma(X) = 2(WX - B(X)X)$, where the matrix $W = [w_{uv}]$ is defined as

$$w_{uv} = \begin{cases} -\pi_{uv}, & \text{if } u \neq v, \\ \sum_{w \neq u} \pi_{uw} & \text{otherwise,} \end{cases}$$

and $B(X) = [b_{uv}(X)]$ is a function of X and is defined as

$$b_{uv}(X) = \begin{cases} -\frac{\pi_{uv} \delta_{uv}}{\|x_u - x_v\|}, & \text{if } u \neq v \text{ and } \|x_u - x_v\| > 0, \\ 0, & \text{if } u \neq v \text{ and } \|x_u - x_v\| = 0, \\ -\sum_{w \neq u} b_{uw}(X), & \text{otherwise.} \end{cases}$$

In [6], Glunt et al. have proposed a pure spectral gradient method for the solution of this optimization problem. This method does not require line search, but convergence was established only for strictly convex quadratic functions. In order to ensure global convergence from arbitrary starting points, we consider the non-monotone line search strategy initially proposed by Grippo, Lampariello and Lucidi in [7] and subsequently by Zhang and Hager in [16]. Although the gradient $\nabla \sigma(X)$ may not be continuous when $\|x_u - x_v\| = 0$, we did not observe numerical instabilities in our computational experiments within the required tolerance in the stopping criteria.

3 A non-monotone spectral gradient method

For a general unconstrained optimization problem, where f is a continuously differentiable objective function, iterative methods produce a sequence of points $\{X_k\}_{k \in \mathbb{N}}$ where each X_{k+1} is generated from X_k by applying the formula $X_{k+1} = X_k + \alpha_k D_k$, where D_k gives the search direction in the domain of the objective function, and α_k is the step that is performed from X_k in the direction D_k . Since $-\nabla f(X_k)$ provides the direction

Algorithm 1 The non-monotone spectral gradient method.

```

1: NonmonotoneSpectralGradient ( $X_0, \mu_{\min}, \mu_{\max}, \varepsilon, \gamma, \eta_k$ )
2: set  $k = 0, Q_0 = 1, C_0 = \sigma(X_0)$ ;
3: while ( $\|\nabla\sigma(X)\| > \varepsilon$ ) do
4:   evaluate  $\sigma(X_k)$  and  $\nabla\sigma(X_k)$ ;
5:   if ( $k = 0$ ) then
6:     set  $D_0 = -\nabla\sigma(X_0)$ ; go to line 11;
7:   end if
8:   let  $Y_{k-1} = \nabla\sigma(X_k) - \nabla\sigma(X_{k-1})$ ; let  $S_{k-1} = X_k - X_{k-1}$ ;
9:   let  $\mu_k = \min\left(\mu_{\max}, \max\left(\mu_{\min}, \frac{\langle Y_{k-1}, S_{k-1} \rangle}{\langle S_{k-1}, S_{k-1} \rangle}\right)\right)$ ;
10:  let  $D_k = -\frac{1}{\mu_k} \nabla\sigma(X_k)$ ;
11:  let  $\alpha_k = 1$ ;
12:  while ( $\sigma(X_k + \alpha_k D_k) > C_k + \gamma\alpha_k \langle \nabla\sigma(X_k), D_k \rangle$ ) do
13:    let  $\alpha_k = \alpha_k/2$ ;
14:  end while
15:  let  $X_{k+1} = X_k + \alpha_k D_k$ ; let  $Q_{k+1} = \eta_k Q_k + 1$ ; let  $C_{k+1} = (\eta_k Q_k C_k + \sigma(X_{k+1}))/Q_{k+1}$ ;
16:  let  $k = k + 1$ ;
17: end while

```

of steepest decrease of the objective function from X_k , it is common that D_k is proportional to $-\nabla f(X_k)$. When f is twice continuously differentiable, Newton-type methods use directions of the form $D_k = -H_k^{-1} \nabla f(X_k)$, where H_k is an approximation for the Hessian of f [14].

Since the work of Barzilai and Borwein [1], the use of spectral gradient methods have been employed with success in large scale optimization [3]. In such methods, $H_k = \mu_k I$, for $\mu_k > 0$, where I is the identity matrix. As proposed in [1, 6], we adopt $\mu_k = \langle Y_{k-1}, S_{k-1} \rangle / \langle S_{k-1}, S_{k-1} \rangle$, where $Y_{k-1} = \nabla f(X_k) - \nabla f(X_{k-1})$, $S_{k-1} = X_k - X_{k-1}$ and $\langle A, B \rangle$ denotes the inner product, computed as the trace of $B^\top A$.

The choice of the step α_k is crucial in a line search: when it is too short, further improvements may be possible on the objective function value; when it is too long, the best objective function values may be missed during the step. In monotone searches, it is imposed that $f(X_{k+1}) < f(X_k)$, for every k . However, it seems reasonable to combine a non-monotone line search strategy with a spectral gradient method, because in the latter the objective function does not generally decrease monotonically.

For the optimization problem presented in Section 2, for every sub-instance $G[V \times \{t\}]$, we employed the spectral gradient method with the non-monotone line search method proposed by Zhang and Hager [16]. The main steps are sketched in Alg. 1. For this particular line search, global convergence was proved for smooth and non-convex functions, and R -linear convergence was proved for strongly convex functions.

In Alg. 1, we have used the classical choice for μ_k with safeguards $\mu_{\min} = 10^{-4}$ and $\mu_{\max} = 10^4$ in order to guarantee a positive bounded sequence $\{\mu_k\}$. The iterations are stopped when $\|\nabla\sigma(X_k)\| < \varepsilon = 10^{-8}$. At line 11, the algorithm attempts to perform an initial step $\alpha = 1$. While the non-monotone Armijo condition is not satisfied (with $\gamma = 10^{-4}$), the value of α is divided by 2. Following the non-monotone line search in [16], by setting $\eta_k = 0$ one obtains a classical monotone line-search whereas $\eta_k = 1$

(used in the our experiments) implies a non-monotone line search where C_k corresponds to the average of objective function values over all previous iterations.

Concerning the starting point, for $t = 1$, we randomly select the vertex positions for X_0 and center this realization around the origin. For all other $t > 0$, X_0 is set to the solution obtained for the sub-instance $G[V \times \{t - 1\}]$. One interesting remark about Alg. 1 is that if X_0 is centered, then all iterates are centered due to the forms of $B(X)$ and W [6]. For more information about the algorithm, the reader is referred to the publications cited above.

4 Computational experiments

We present some computational experiments on a set of artificially generated instances of the dynDGP. All codes were written in Matlab 2016b and the experiments were carried out on an Intel Core 2 Duo @ 2.4 GHz with 2GB RAM, running Mac OS X.

We mainly focus on two kinds of motions, both defined in a two-dimensional Euclidean space. The *cascade* motion is an animation consisting of 8 points initially placed at the top of a $[0, 1] \times [0, 1]$ domain and subsequently moving downwards in the direction of the x axis in a cascade fashion. Two of such points are fixed during the motion, in order to avoid to define frames with almost constant inter-point distances. We refer to the second motion that we consider as the *black-hole* motion. Only one vertex remains steady in the center of a $[0, 1] \times [0, 1]$ domain, while the others, initially located around the borders of the domain, tend to come closer and closer to the steady vertex. All considered motions are composed by 100 frames. The reader can make reference to the first and third column in Fig. 1 for having a better understanding of the described motions.

The two motions were initially generated by defining the trajectories of the coordinates of their 8 vertices in the two-dimensional space. From these trajectories, we generated two instances of the dynDGP such that the corresponding graph G has an edge set E for which $\hat{E} = \emptyset$. We were able to solve the inverse problem, i.e. the one of reconstructing the trajectories from the distance information in G , by solving the optimization problem presented in Section 2, with the method described in Section 3. The quality of the found solutions is measured with the function

$$MDE(X) = \sum_{\{u,v\} \in E_t} \frac{||x_u - x_v|| - \delta_{uv}}{\delta_{uv}},$$

which has maximal values $4 \cdot 10^{-6}$ and $8 \cdot 10^{-10}$, respectively, for the cascade and black-hole motions (see Fig. 1, first and third columns).

We also consider instances of the dynDGP with modified distances. Given a graph G obtained from a known motion, we wish to control this motion by imposing a new set of distance constraints. In our cascade motion, we impose that the distance δ'_{uv} between two particular vertices increases by a factor $\tau \in [0, 3]$ during the motion ($\delta'_{uv} \leftarrow \tau \delta_{uv}$, for all t). In the black-hole motion, we impose instead that the distance δ'_{uv} between the steady vertex and any other is always greater than a given threshold $d_0 > 0$ ($\delta'_{uv} \leftarrow \max(d_0, \delta_{uv}^t)$).

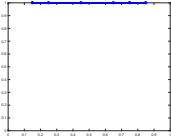
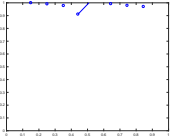
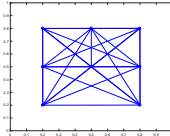
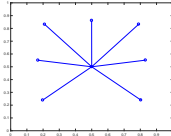
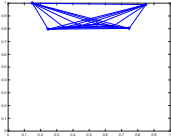
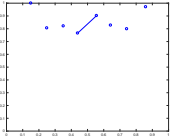
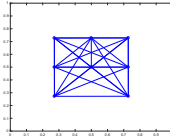
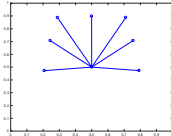
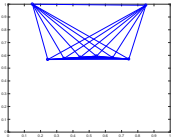
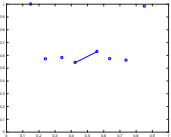
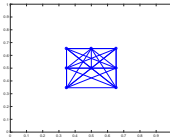
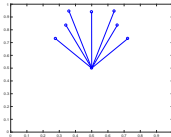
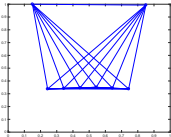
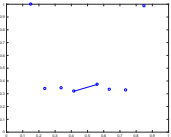
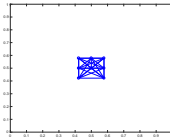
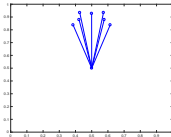
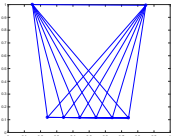
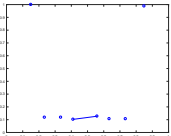
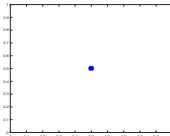
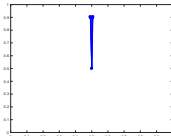
cascade		black-hole	
original	modified	original	modified
$t = 1$  MDE: 8.4718e-06	$t = 1$  MDE: 0.0322	$t = 1$  MDE: 8.0183e-10	$t = 1$  MDE: 0.0503
$t = 25$  MDE: 4.8354e-08	$t = 25$  MDE: 0.0396	$t = 25$  MDE: 1.0404e-13	$t = 25$  MDE: 0.0989
$t = 50$  MDE: 1.5189e-08	$t = 50$  MDE: 0.0399	$t = 50$  MDE: 1.5175e-16	$t = 50$  MDE: 0.1479
$t = 75$  MDE: 9.1084e-09	$t = 75$  MDE: 0.0472	$t = 75$  MDE: 2.6909e-16	$t = 75$  MDE: 0.1499
$t = 99$  MDE: 1.0540e-08	$t = 99$  MDE: 0.0503	$t = 99$  MDE: 1.4821e-15	$t = 99$  MDE: 0.1340

Fig. 1. Some selected frames of the obtained motions, for both instances cascade and black-hole, without and with some modifications on the original distances ($\tau = 3.0$ and $d_0 = 0.4$). The segments mark the distances with highest priorities. When no distances are modified, they all have the same priority.

<i>cascade</i>			<i>black-hole</i>		
τ	MDE_{min}	MDE_{max}	d_0	MDE_{min}	MDE_{max}
1.0	2.4035e-09	4.5095e-06	0.0	1.0637e-16	8.0183e-10
1.4	0.0113	0.0608	0.1	8.0183e-10	0.1518
1.8	0.0245	0.0416	0.2	8.0183e-10	0.1518
2.0	0.0315	0.0504	0.3	1.7766e-08	0.1518
3.0	0.0639	0.1016	0.4	0.0503	0.1518

Table 1. Computational experiments with different values for τ and d_0 , for both instances cascade and black-hole. The smallest and the largest MDE values over the frames at time t of the obtained motions are reported.

In both cases, these modifications on the original distances concern a small percentage of distances. Therefore, the optimization process may tend to optimize the original and “genuine” distances while ignoring the new ones, because they would only increase a relatively small number of terms in the objective function. The use of the weights π_{uv}^t on the distances δ_{uv}^t becomes therefore fundamental: in our experiments with modified distances, we give two levels of priorities to the distances. Higher priority is given to the modified distances, i.e. $\pi_{uv}^t = 1$ (π_{uv}^t is instead set to 0.5 when the corresponding distance value is not modified).

Table 1 shows the smallest and the largest values for the MDE function for all frames t composing the obtained motions. This table shows an increase of the MDE values when the values of τ and d_0 are larger: this was expected because the perturbation on the original distances becomes more and more significant. It is important to remark that the MDE values may become smaller when assigning no priorities to the distances, but then, as mentioned above, the obtained motions may not satisfy the newly introduced distance constraints. In this context, the analysis of the motion by a viewer, rather than the measure of a quality index such as the MDE, is often preferred for validating the results (see Fig. 1, second and forth columns).

5 Conclusions

We introduced the dynDGP where an embedding represents a motion of a given set of objects that satisfies the distance constraints in the graph G . We focused our attention on problems where no inter-frame distance information is given, and proposed to formulate an optimization problem for the identification of every frame of the motion, where the distance values are coupled with priority levels.

This work is motivated by various emerging real-life applications. In character animation, for example, motions are generally simulated by modifying pre-recorded motions [13], and an attempt to perform this modification through distance constraints was already proposed in [9]. In air-traffic control, the positions of a set of flying airplanes needs to be predicted so that some distance constraints are satisfied, which are defined for guaranteeing collision avoidance (see for example the recent work in [15]). Similarly, in crowd simulations, one is interested in simulating a crowd motion in different

situations, where distances between pairs of pedestrians can be estimated and exploited for the simulations [4].

The formulation of these applications as a dynDGP, as well as the applicability of the proposed method for the computation of the motions, will be the subject of future research. Notice that our assumption $\hat{E} = \emptyset$ may not be feasible for all these applications: our method may need to be extended and adapted to the different situations.

Acknowledgments

This work was partially supported by an INS2I-CNRS 2016 “PEPS” project. The authors are thankful to Franck Multon and Ludovic Hoyet for the fruitful discussions.

References

1. J. Barzilai, J. Borwein, *Two-Point Step Size Gradient Methods*, IMA Journal of Numerical Analysis **8**, 141–148, 1988.
2. E.G. Birgin, J.M. Martínez, M. Raydan, *Nonmonotone Spectral Projected Gradient Methods on Convex Sets*, SIAM Journal on Optimization **10**, 1196–1211, 2000.
3. E.G. Birgin, J.M. Martínez, *Large-Scale Active-Set Box-Constrained Optimization Method with Spectral Projected Gradients*, Computational Optimization and Applications **23**, 101–125, 2002.
4. J. Bruneau, T.B. Dutra, J. Pettré, *Following Behaviors: a Model for Computing Following Distances*, Transportation Research Procedia **2**, 424–429, 2014.
5. J. de Leeuw, *Convergence of the Majorization Method for Multidimensional Scaling*, Journal of Classification **5**, 163–180, 1988.
6. W. Glunt, T.L. Hayden, M. Raydan, *Molecular Conformations from Distance Matrices*, Journal of Computational Chemistry **14**(1), 114–120, 1993.
7. L. Grippo, F. Lampariello, S. Lucidi, *A Nonmonotone Line Search Technique for Newton’s Method*, SIAM Journal on Numerical Analysis **23**, 707–716, 1986.
8. L. Grippo, F. Lampariello, S. Lucidi, *A Truncated Newton Method with Nonmonotone Line Search for Unconstrained Optimization*, Journal of Optimization Theory and Applications **60**, 401–419, 1989.
9. Th. Le Naour, N. Courty, S. Gibet, *Cinématique guidée par les Distances*, Revue Electronique Francophone d’Informatique Graphique, Association Française d’Informatique Graphique **6**(1), 15–25, 2012.
10. L. Liberti, C. Lavor, N. Maculan, A. Mucherino, *Euclidean Distance Geometry and Applications*, SIAM Review **56**(1), 3–69, 2014.
11. A. Mucherino, R. de Freitas, C. Lavor, *Distance Geometry and Applications*, special issue of Discrete Applied Mathematics **197**, 1–144, 2015.
12. A. Mucherino, C. Lavor, L. Liberti, N. Maculan (Eds.), *Distance Geometry: Theory, Methods and Applications*, 410 pages, Springer, 2013.
13. F. Multon, L. France, M.P. Cani-Gascuel, G. Debunne, *Computer Animation of Human Walking: a Survey*, The Journal of Visualization and Computer Animation **10**(1), 39–54, 1999.
14. J. Nocedal, S.J. Wright, *Numerical Optimization*, Springer Series in Operations Research and Financial Engineering, Springer, 2006.
15. J. Omer, *A Space-Discretized Mixed-Integer Linear Model for Air-Conflict Resolution with Speed and Heading Maneuvers*, Computers and Operations Research **58**, 75–86, 2015.
16. H. Zhang, W.W. Hager, *A Nonmonotone Line Search Technique and its Applications to Unconstrained Optimization*, SIAM Journal of Optimization **14**(4), 1043–1056, 2004.